Physica A xx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Directed networks' different link formation mechanisms causing degree distribution distinction

Stefan Kambiz Behfar [a,b,c,*], Ekaterina Turkina [d], Patrick Cohendet [d], Thierry Burger-Helmchen [c]

[a] Chair of Entrepreneurial Risk, Department of Management, Technology and Economics, ETH Zurich, Switzerland
[b] Institute for Solid State Physics, Department of Physics, ETH Zurich, Switzerland
[c] BETA — CNRS, University of Strasbourg, France
[d] Department of International Business, HEC Montreal, Canada

HIGHLIGHTS

- Outlinks feature different degree distributions than inlinks.
- Different link formation mechanisms cause the degree distribution distinctions.
- In/outdegree distribution distinction holds for different levels of system decomposition.

ARTICLE INFO

Article history:
Received 7 May 2015
Received in revised form 31 March 2016
Available online xxxx

Keywords:
Directed Network
Link formation mechanism
Inlink and Outlink
Complex Network
Open-Source-Software (OSS)
System decomposition level

ABSTRACT

Within undirected networks, scientists have shown much interest in presenting power-law features within complex networks. For instance, Barabási and Albert (1999) claimed that a common property of many large networks was that vertex connectivity follows scale-free power-law distribution, and in another study Barabási et al. (2002) showed power law evolution in the social network of scientific collaboration. At the same time, Jiang et al. (2011) discussed deviation from power-law distribution; others indicated that size effect (Bagrow et al., 2008), information filtering mechanism (Mossa et al., 2002), and birth and death process (Shi et al., 2005) could account for this deviation. Within directed networks, many authors have considered that outlinks follow a similar mechanism of creation as inlinks' formation (Faloutsos et al., 1999; Krapivsky et al., 2001; Tanimoto, 2009) with link creation rate being the linear function of node degree, and a resulting power-law shape for both indegree and outdegree distribution. Some other authors have made an assumption that directed networks, such as scientific collaboration or citation, behave as undirected, resulting in a power-law degree distribution accordingly (Barabási et al., 2002). At the same time, we claim (1) Outlinks feature different degree distributions from inlinks; where different link formation mechanisms cause the distribution distinctions, (2) in/outdegree distribution distinction holds for different levels of system decomposition; therefore this distribution distinction is a property of directed networks. First, we emphasize in/outlink formation mechanisms as causal factors for distinction between indegree and outdegree distributions (where this distinction has already been noticed in Barker et al. (2010) and Baxter et al. (2006)) within a sample network of OSS projects as well as Java software corpus as a network. Second, we analyze whether this distribution distinction holds for different levels of system decomposition: open-source-software (OSS) project–project dependency within a

* Corresponding author at: Chair of Entrepreneurial Risk, Department of Management, Technology and Economics, ETH Zurich, Switzerland.
E-mail addresses: behfarka@phys.ethz.ch, kambiz.behfar@gmail.com (S.K. Behfar).

http://dx.doi.org/10.1016/j.physa.2016.06.035

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035

cluster, package–package dependency within a project and class–class dependency within a package. We conclude that indegree and outdegree dependencies do not lead to similar type of degree distributions, implying that indegree dependencies follow overall power-law distribution (or power-law with flat-top or exponential cut-off in some cases), while outdegree dependencies do not follow heavy-tailed distribution.

1. Introduction

Q3

Among network models Erdős–Rényi (ER) [1] proposed a non-growing randomly connected model, Watts and Strogatz (WS) [2] proposed a non-growing randomly re-connected network model (so called small world) and Barabási–Albert (BA) [3] proposed a growing network with the probability of addition of new nodes proportional to the number of incoming links (so-called preferential attachment model or rich-get-richer). In ER and WS models, number of nodes in the network is fixed, and linkages among existing link formation nodes are formed, while BA model assumes time-homogeneous network growth with a mechanism for preferential attachment link formation. There are also other growth models such as fitness model (Bianconi et al. [4]) attractiveness model (Dorogovtesev et al. [5]), accelerating growth model (Dorogovtesev et al. [6]), logarithmic growth model (Shi et al. [7]), and random preferential attachment model (Liu et al. [8]).

Preferential attachment does not always explain network evolution, e.g. where the innovation of an article rather than the number of its citations causes a new attachment. Scientists such as Ergun et al. [9] and Xu et al. [10] have discussed a methodology of fit-get-richer, implying that new vertices connect to highly fitted vertices. This explains attachment to a new network based on its intrinsic physical property or quality. In this area, Caldarelli et al. [11] introduced a varying vertex fitness model. As far as link formation mechanisms are concerned, Newman [12] defined assortativity mixing for *undirected* networks as a node tendency to connect to other nodes with similar degree. Piraveenan [13,14] defined this for *directed* networks as: in (out)-assortativity is the tendency whereby nodes tend to connect to other nodes with similar in (out)-degrees. Jackson and Rogers [15] have also presented a dynamic model of link formation based on random as well as searching through the current structure.

Scientists have shown much interest in presenting power-law features within complex networks. For instance, Barabási and Albert [3] claimed a common property of many large networks was that vertex connectivity follows scale-free power-law distribution, and concluded that development of large networks is governed by robust self-organizing phenomena that go beyond the particulars of the individual systems. Barabási et al. [16] have also shown power law evolution in the social network of scientific collaboration. Furthermore, Faloutsos et al. [17] showed power-law features existing in the internet topology, implying its benefits in designing efficient protocols, creating accurate artificial models and speculating on the internet topology in the future.

Authors such as Jiang et al. [18] discussed a network model of *deviation* from power-law distribution. Some other authors had previously addressed this deviation and indicated that size effect (Bagrow et al. [19]), information filtering mechanism (Mossa et al. [20]), and birth and death process (Shi et al. [21]) accounted for this deviation. Maillart et al. [22] tested Zipf's degree distribution via link creation and deletion mechanism in open source Linux distribution. In another work, Maillart et al. [23] used data collected by Google to identify the existence of power-law regimes for a population of Internet users to execute a given task after receiving a message.

We argue that WWW, Scientific Collaboration and OSS reuse networks are not undirected, as assumed in some studies; they are in fact directed networks where outlinks and inlinks demonstrate different degree distributions. Faloutsos et al. [17], Krapivsky et al. [24], Tanimoto [25] and more also assumed that preferential attachment is the dominant link formation mechanism in directed networks, resulting in power-law degree distribution. At the same time, we claim that there are different (out) inlink formation mechanisms within directed networks which result in degree distribution distinctions. We propose two hypotheses to explain causal effect of link formation mechanism on degree distribution.

We prove the hypotheses both analytically and empirically. In the analytical approach, apart from using indegree-based preferential attachment mechanism to prove our claims, we apply other link formation mechanisms such as outdegree-based preferential attachment, fitness-based preferential attachment. In the empirical section, we first consider the sample network of open-source-software (OSS) projects reuse to identify the distinction between indegree and outdegree distribution, then analyze whether this distinction holds in the corpus of each of those OSS projects, and at different system decomposition levels of package–package and class–class dependencies.

2. Theoretical development and hypotheses

2.1. Inlink and outlink formation logic

As already mentioned, inlink and outlink do not lead to similar types of degree distribution. Here we give few examples to demonstrate the logic behind this distinction.

- Complex stock trading network:

Stock trading can be modeled for each transaction day where investors represent nodes and each transaction represents a directed link from seller to buyer, and trading size represents weight of each edge. Jiang et al. [26] showed a power-law distribution in such a trading network. According to Barabási and Albert (BA) [3] the mechanism underlying inlink formation is (degree-based) preferential attachment, which leads to scale-free distribution. In the case of a network of creditors and individuals leveraging, quality-based attachment is the mechanism underlying formation of outlinks from creditors to individuals. Intuitively, this does not lead to heavy-tailed distribution, as creditors choose a finite number of individuals based on selection and their quality.

- Contact network of contagious virus epidemics:

In an epidemic network of respiratory spread agents, outlinks represent the number of infectious contacts produced by individuals. In this case, outdegree will not follow heavy-tailed distribution because the mechanism underlying outlink formation is random (and not preferential attachment), although only vulnerable persons might be affected.

- Webpage-ranking network:

Page ranking considers pages with many incoming links (infinitely) and few outgoing links as sources of information that are governed by indegree-based preferential attachment link formation mechanism; they follow power-law degree distribution. On the other hand, pages with many outgoing links and few incoming links represent portals; the mechanism underlying this outlink formation could either be outdegree-based preferential attachment or quality-based attachment. Degree distribution for these portals will not be heavy-tailed.

- Open Source Software (OSS) network:

When OSS projects call or reuse each other, inlink represents linkage from project A (caller) to project B (called or reused). Popular projects (with highest indegree of reuse) are called far more often than other projects (infinitely), considering that inlink formation mechanism is based on indegree-based preferential attachment, indegree distribution follows power-law. However, large projects call small projects less frequently (finitely), considering that outlink formation attachment mechanism is based on quality-based attachment, degree distribution will probably not follow power-law. In the case of network of OSS project corpus, a software package could be indefinitely reused (indegree dependency). Therefore, there tend to be some packages with very small and some with very large indegrees, therefore its indegree distribution is heavy-tailed. While a package could r*euse* limited number of other software packages based on their quality (outdegree dependency), so its outdegree distribution tends to not be heavy-tailed.

- Citation network:

Barabási et al. [16] made an assumption that scientific collaboration or paper citation network is undirected, and concluded a power-law degree distribution accordingly, although citation network is in fact a directed network. Barker et al. [27] claimed that power-law degree distribution for citations within a corpus of scholarly articles or web pages appears to be approximately true for incoming citations, whereas there is a substantial deviation from a power law for outgoing citations. In the case of a paper citation network, one popular (more Visible) paper could be indefinitely cited. Therefore, some papers tend to have very small and others very large indegrees, and the network's indegree distribution is heavy-tailed. When a paper cites a limited number of other papers, based on their quality (more Relevant), then its' outdegree distribution tends not to be heavy-tailed.

2.2. Hypotheses

Barabási et al. [16] inferred the dynamic and the structural mechanisms that govern the evolution of a co-authorship network. This network of scientists represents a prototype of a complex social network through mapping the electronic database containing all relevant journals in mathematics (M) and neuro-science (NS) for an eight year period (1991–1998). Barabási showed that degree distribution for both Math and Neuroscience exhibits power law distribution. On the other hand, Barker et al. [27] claimed that power-law degree distribution for citations within a corpus of scholarly articles or web pages appears to be approximately true for incoming citations, whereas there is a substantial deviation from a power law for outgoing citations. Similarly, we argue that WWW, Scientific Collaboration and OSS reuse networks are not undirected, as assumed in some studies; they are in fact directed networks where outlinks and inlinks demonstrate different degree distributions. Faloutsos et al. [17], Krapivsky et al. [24], Tanimoto [25] and more also assumed that preferential attachment (link creation rate as linear function of node degree) is the dominant link formation mechanism in directed networks, resulting in power-law degree distribution. At the same time, we claim that there are different (out) inlink formation mechanisms within directed networks which result in degree distribution distinctions. Moreover, we propose a hypothesis to explain causal effect of link formation mechanism on degree distribution as:

Hypothesis 1. Outlinks feature different degree distributions from inlinks; and different link formation mechanisms cause the distribution distinctions.

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035

There are some properties of complex networks, where knowing degree distribution type is of great significance.

- Structural property of complex networks

For the purpose of design it is important to know what the network looks like. For instance, as mentioned by Baxter et al. [28], we believe that current methodologies for software design lead to reusable, testable good software, but without knowing what good software resembles, how can we know if these methodologies really work?

- Statistical property of complex networks

We should know the type of degree distribution for the underlying complex network. For instance, if power-law distribution may not have finite mean and variance, then central limit theorem does not apply, and therefore sample mean and variance cannot be used as the estimator of the population mean and variance.

- Self-organizing property of complex networks

Scale-free distribution has self-organizing property. Barabási and Albert [3] claimed common property of many large networks in that the vertex connectivity follows scale-free distribution, and concluded that development of large networks is governed by robust self-organizing phenomena beyond the particulars of individual systems.

- Distinctive property of directed networks

In this study, we consider sample network of OSS project reuse; we show that there is distinction between indegree and outdegree distribution. Also we consider sample network of java software corpus, where one also observes distinction between indegree and outdegree distributions. Therefore, this distribution distinction holds for different levels of system decomposition. Simon [29] argued that complex systems more generally including biological and computing systems can be decomposed into constituent parts that operate in relative isolation from each other. Here system components are 1. Cluster of projects, 2. projects (including packages), 3. packages (including classes); and system interactions are 1. project–project dependency within a cluster, 2. package–package dependency within a project, 3. class–class dependency within a package. Considering that complex systems can be decomposed into different levels, we would like to show that *in/outdegree distribution distinction should exist.*

Hypothesis 2. In/outdegree distribution distinction holds for different levels of system decomposition; therefore this distribution distinction is a property of directed networks.

2.3. Analytical approach to link formation mechanisms

2.3.1. InLink formation mechanisms

Barabási–Albert (BA) [3] proposed a growing network with the probability of addition of new nodes proportional to the number of incoming links (indegree-based preferential attachment). There are also other inlink formation mechanisms, as listed below:

1. Indegree-based preferential attachment
2. Outdegree-based preferential attachment
3. Preferential attachment fitness model.

Although most studies define preferential attachment as: one becomes popular because of receiving many inlinks; however, one could also become popular because of having many outlinks, e.g. hub gamers outlinking to many other teams, web portals outlinking to many other websites and more; we call this out-degree preferential attachment.

1. Indegree-based preferential attachment mechanism implies that the probability of a new node inlinked to node i depends on the indegree k_i^{in} of that node such that $\varphi\left(k_i^{in}\right) = k_i^{in} / \sum_j k_j^{in}$. The indegree distribution $P(k_i^{in} < k)$ can be obtained below, as shown by Barabási et al. [30]. If number of initial nodes is denoted by m_0, after t time steps, network includes $N = m_0 + t$ nodes and mt links. Then continuous rate of change of k_i^{in} is as:

$$\frac{\partial k_i^{in}}{\partial t} = \mu_1 m \varphi\left(k_i^{in}\right) = \mu_1 m \frac{k_i^{in}}{\sum_{j=1}^{m_0+t-1} k_j^{in}}. \tag{1}$$

Take into account m links added to the network at each time step, after t steps the total quantity of degree increase is $\sum_j k_j^{in} = mt$, where $\mu_1 m$ is the quantity of degree increase for part of new nodes preferentially attached to already existing nodes, and $(1 - \mu_1)m$ represents degree increase for new nodes at each time step (Tanimoto [25]).

$$\frac{\partial k_i^{in}}{\partial t} = \mu_1 m \frac{k_i^{in}}{mt} = \mu_1 \frac{k_i^{in}}{t}. \tag{2}$$

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035

Considering the initial condition for connectivity of node added to network $k_i^{in}(t_i) = (1-\mu_1)m$, then

$$k_i^{in}(t) = (1-\mu_1)m\left(\frac{t}{t_i}\right)^{\mu_1} \quad t \gg t_i. \tag{3}$$

Older vertices with smaller t_i increase their connectivity at expense of younger vertices with larger t_i; this results in highly-connected vertices. These phenomena help us calculate $P(k_i^{in} < k)$.

$$P(k_i^{in} < k) = P\left(t_i > \frac{t\,(m(1-\mu_1))^{1/\mu_1}}{k^{1/\mu_1}}\right). \tag{4}$$

Assuming that new vertices are added in equal time step $P_i(t_i) = 1/(m_0+t)$, then

$$P\left(t_i > \frac{t\,(m(1-\mu_1))^{1/\mu_1}}{k^{1/\mu_1}}\right) = 1 - P\left(t_i < \frac{t\,(m(1-\mu_1))^{1/\mu_1}}{k^{1/\mu_1}}\right) = 1 - \frac{t\,(m(1-\mu_1)^{1/\mu_1}}{(t+m_0)k^{1/\mu_1}}. \tag{5}$$

Density will be power law: $p(k) \sim k^{-(1+\frac{1}{\mu_1})}$

$$p(k) = \frac{\partial P(k_i^{in}(t) < k)}{\partial k} = \frac{t\,(m\,(1-\mu_1))^{\frac{1}{\mu_1}}}{\mu_1(t+m_0)}k^{-\left(1+\frac{1}{\mu_1}\right)}. \tag{6}$$

2. Outdegree-based preferential attachment mechanism implies that the probability of a new node inlinked to node i depends on the outdegree k_i^{out} such that $\varphi\left(k_i^{in}\right) = k_i^{out}/\sum_j k_j^{out}$. Take into account m links added to the network at each time step, after t steps the total quantity of degree increase is $\sum_j k_j^{in} = mt$, where $\mu_2 m$ is the quantity of degree increase for part of new nodes preferentially attached to already existing nodes, and $(1-\mu_2)m$ represents degree increase for new nodes at each time step (Tanimoto [25]).

$$k_i^{out}(t) = (1-\mu_2)m\left(\frac{t}{t_i}\right)^{\mu_2} \quad t \gg t_i \tag{7}$$

$$\frac{\partial k_i^{in}}{\partial t} = \mu_1 m\frac{k_i^{out}}{mt} = \mu_1\frac{(1-\mu_2)\,m\left(\frac{t}{t_i}\right)^{\mu_2}}{t}. \tag{8}$$

Consider initial condition $k_i^{out}(t_i) = (1-\mu_2)m$

$$k_i^{in} = \frac{\mu_1(1-\mu_2)\,m}{\mu_2}\left(\frac{t}{t_i}\right)^{\mu_2} + \frac{(\mu_2-\mu_1)\,m}{\mu_2}. \tag{9}$$

These help us calculate $P(k_i^{in} < k)$

$$P(k_i^{in} < k) = P\left(t_i > \left(\frac{\mu_1(1-\mu_2)\,m}{(\mu_1-\mu_2)\,m + k\mu_2}\right)^{\frac{1}{\mu_2}} t\right). \tag{10}$$

Assuming that new vertices are added in equal time step $P_i(t_i) = 1/(m_0+t)$, then,

$$P\left(t_i > \left(\frac{\mu_1(1-\mu_2)\,m}{(\mu_1-\mu_2)\,m + k\mu_2}\right)^{\frac{1}{\mu_2}} t\right) = 1 - P\left(t_i < \left(\frac{\mu_1(1-\mu_2)\,m}{(\mu_1-\mu_2)\,m + k\mu_2}\right)^{\frac{1}{\mu_2}} t\right)$$

$$= 1 - \left(\frac{\mu_1(1-\mu_2)\,m}{(\mu_1-\mu_2)\,m + k\mu_2}\right)^{\frac{1}{\mu_2}}\frac{t}{(t+m_0)}. \tag{11}$$

Density will be a power law, if $\mu_1 = \mu_2$.

$$p(k) = \frac{\partial P(k_i^{in}(t) < k)}{\partial k} = \frac{\mu_2\,t}{(t+m_0)}\left(\frac{1}{(\mu_1-\mu_2)\,m + k\mu_2}\right)^{\frac{1}{\mu_2}}. \tag{12}$$

3. Fitness model implies fit-get-richer instead of rich-get-richer (BA preferential attachment model). The problem with BA model is that for instance in case of citation network, BA model does not allow for a very good scientific paper to be more cited than older but less important one. A new network evolution model has been proposed (Bianconi et al. [4]), which is based on characteristic of vertex, so-called fitness, in which each node is assigned a fitness x_i from a given probability distribution $f(x)$. The probability that a new node connects with the already-present node i depends on its connectivity

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035

k_i and its fitness x_i.

$$\frac{\partial k_i^{in}}{\partial t} = m\frac{x_i k_i^{in}}{\sum_j x_j k_j^{in}}. \tag{13}$$

Similar to BA model, the time evolution of k_i^{in} follows a power-law. The degree distribution can be obtained by:

$$k_{x_i}^{in}(t) = m\left(\frac{t}{t_i}\right)^{\mu}. \tag{14}$$

Following the steps in degree distribution given by Bianconi et al. [4], $P(k)$ will also follow a power-law.

$$P(k) \propto \int \mathrm{d}x f(x)\ \mu\left(\frac{m}{k}\right)^{1+\mu} \sim \frac{k^{-(1+\mu)}}{\log(k)}. \tag{15}$$

2.3.2. OutLink formation mechanisms

As already explained, outlinks do not follow similar mechanism of creation as inlinks'. Degree-based preferential attachment does not hold for outlink formation underlying mechanism. Large firms outsource to small firms due to business quality, or respiratory spread agents outcontact individuals randomly, or a country decides to choose another country to trade with based on assortativity. Therefore, the outlink formation mechanisms are:

1. Random attachment
2. Quality-based attachment
3. Assortativity.

1. Assuming that each node outlinks randomly to other nodes with same probability within a growing network, where number of initial nodes is denoted by m_0, after t time steps, network includes $N = m_0 + t$ nodes. Then rate of change of k_i^{out} is as:

$$\frac{\partial k_i^{out}}{\partial t} = \mu_2\frac{1}{m_0 + t - 1}. \tag{16}$$

Consider the initial condition $k_i^{out}(t_i) = 0$, at each time step Δk_i^{out} links will be added to the network, and $k_i^{out}(t)$ will be determined as:

$$k_i^{out} = \mu_2 \ln\frac{m_0 + t - 1}{m_0 + t_i - 1}. \tag{17}$$

Then,

$$P(k_i^{out} < k) = P\left(t_i > (m_0 + t - 1)\exp\left(-\frac{k}{\mu_2}\right) - m_0 + 1\right) = 1 - \frac{(m_0 + t - 1)\exp\left(-\frac{k}{\mu_2}\right) - m_0 + 1}{m_0 + t}. \tag{18}$$

Density shows exponential distribution for random link formation.

$$p(k) = \frac{\partial P(k_i^{out}(t) < k)}{\partial k} = \frac{(m_0 + t) - 1}{\mu_2(m_0 + t)}\mathrm{e}^{\left(-\frac{k}{\mu_2}\right)}. \tag{19}$$

2. In the case of Quality-based attachment, similar to the intrinsic fitness model, the probability of a couple of nodes i, j connecting is $f(x_i, x_j)$, where each node is assigned a fitness x_i (Caldarelli et al. [11]). As opposed to fitness model proposed by Bianconi et al. [4], the preferential attachment rule is eliminated, then:

$$k(x) = N\int_0^{\infty} f(x, y)\, g(y)\, \mathrm{d}y = N F(x). \tag{20}$$

Assuming that $F(x)$ is a monotone function, for large number of nodes,

$$P(k) = f\left[F^{-1}\left(\frac{k}{N}\right)\right]\frac{\mathrm{d}}{\mathrm{d}k}F^{-1}\left(\frac{k}{N}\right). \tag{21}$$

As an example, consider $f(x_i, x_j) = (x_i, x_j)/x_{Max}$, where x_{Max} is the largest value of x in the network, then

$$P(k) = \frac{x_{Max}^2}{N\langle x\rangle}f\left(\frac{x_{Max}^2}{N\langle x\rangle}k\right). \tag{22}$$

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035

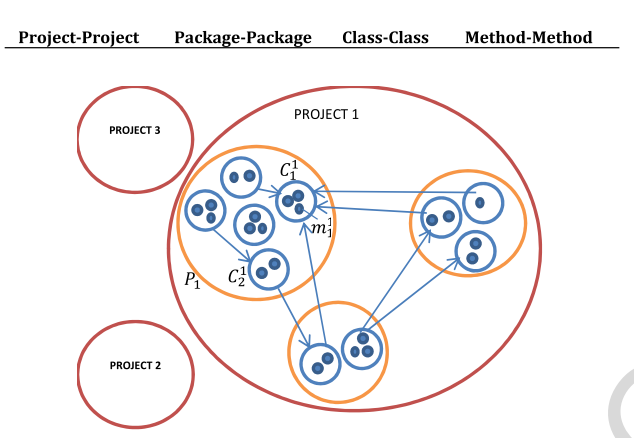

Fig. 1. Illustration of different system decomposition levels in OSS project network, project–project, package–package (P_1), class–class (C_1), method–method (m_1).

If intrinsic fitness function $f(x)$ follows an exponential distribution, then outdegree distribution $P(k)$ will also be exponential. If intrinsic fitness function is Bernoulli, then outdegree $P(k)$ will also be Binomial. As opposed to preferential attachment fitness model for k_i^{in}, we did not assume power-law time evolution for k_i^{out}.

3. Assortativity is also considered under the category of quality-based attachment when fitness of nodes $x_i = x_j$.

3. Empirical approach

In this section, we would like to accomplish numerical proof to distinction between indegree and outdegree distribution in different levels of system decomposition, as illustrated in Fig. 1. We first consider sample network of open-source-software (OSS) projects reuse to identify the distinction between indegree and outdegree distribution for projects illustrated in Figs. 2 and 4, then analyze whether this distinction holds in the corpus of each of those OSS projects, and different system decomposition levels of package–package and class–class dependencies.

3.1. Sample network of OSS projects

In order to build a sample network of OSS projects, using snowball sampling, we adopted one OSS project called "Jpox", subsequently obtained all the projects which reuse "Jpox". Then in the next step, we obtained other projects which reuse those projects that reuse "Jpox" in the first place. We observed that the projects reusing "Jpox" also reuse each other and this also holds in the second level, those projects also reuse each other. We see the network of these projects in Fig. 2, where blue lines represent the simple reuse and red lines show reciprocal reuse.

In the higher level of system decomposition, the sample network is divided into three connected clusters, shown by redlines. In the next step, to distinguish these three clusters, we omit a few low-weighted links (less number of reuse) from Fig. 2. The resulting figure shown in Fig. 4 indicates that: First, the sample OSS network is divided into three connected clusters (numbers inside each circle show clustering coefficient of each project; projects within one cluster have higher clustering coefficients compared to projects outside cluster). Second, these three clusters are mostly connected via "Jpox" and "Data Nucleus" projects. As one observes in the higher level of system decomposition, clusters are also weakly inter-connected, and strongly and reciprocally intra-connected. The local clustering coefficient for a module is given by the proportion of links between the modules within its neighborhood divided by the number of links that could possibly exist between them. For a directed graph, for each neighborhood there are $k(k-1)$ links that could exist among the modules within the neighborhood, while it is equal to $k(k-1)/2$ for undirected network, where k denotes degree of each module. We do not further discuss degree distribution of inlinks and outlinks between clusters.

3.2. Network of OSS java project corpus

In recent years, many studies analyzed software systems from the perspective of complex networks (Baxter et al. [28]), software structures and architecture (Harrison et al. [31]) (Wang et al. [32]) World-Wide-Web and Cellular networks (Ma et al. [33]), and evolution and growth of software dependency networks (Wen et al. [34]). Studying the control properties of complex networks providing insight into how designers and engineers can influence these systems to achieve a desired behavior (Ruths et al. [35]). Exploring software systems and managing dependencies reflect design and implementation of the underlying system. Managing dependencies is useful for programmers to evaluate the impact of a change, and is useful for reviewers and architects for assessing the coupling within an application. We consider OSS java project corpus as a

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035

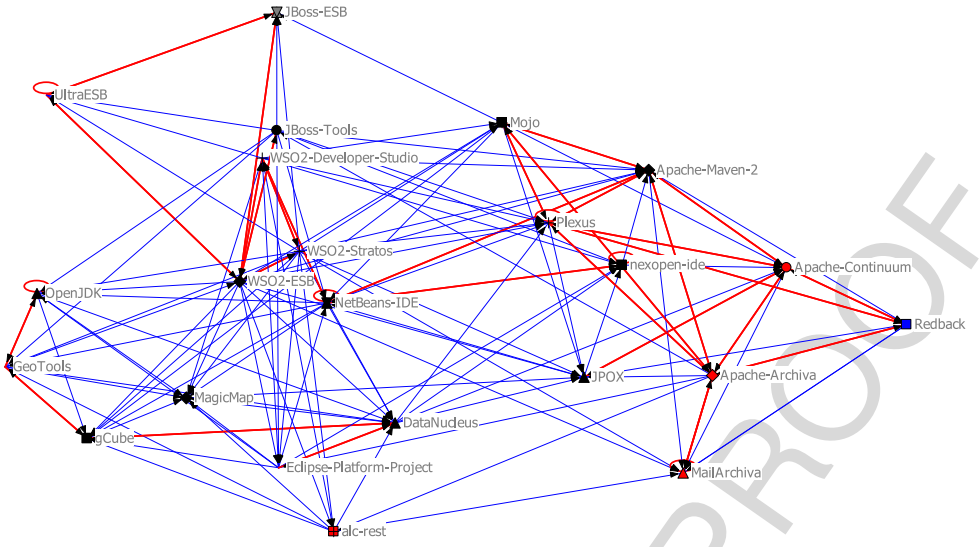

Fig. 2. Illustration of sample network of OSS software projects, where blue lines show the simple reuse and red lines show reciprocal reuse. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

prototype for complex network where modules (classes and packages) represent nodes and module dependencies represent links. We use three dependency softwares dependency finder (Depfinder) [36], dependency analyzer (Depanalyzer) [37], and Jdepend [38] to obtain a dependency matrix and then compute out and indegree dependencies for Class–Class dependencies and Package–Package dependencies. We conclude that in/outdegree dependencies do not lead to similar types of distribution. Java Project Corpus Dependency shows very different characteristics including in/outdegree, implicit/explicit, concrete/abstract, at different levels of system decomposition (package, class, method), where these might make distinctions in the degree distributions. We use complex network modeling where dependency graph comprises nodes for software artifacts linked using in/outdegree dependency. We only refer to dependencies for classes and packages, where classes refer to each other, packages call each other, and they constitute two system decomposition levels of software corpus network.

- Class dependencies:

If some classes require other classes to do their operations, the former classes are dependent on the latter classes. Generally, if one element A requires another element B to do its operation, then one is dependent, and the other is dependable.

- Indegree and outdegree dependencies:

It depends how to look at dependency. Say that class A is dependent and class B is dependable. A depends on B ($A \rightarrow B$). We say that A has outdegree dependency while B has indegree dependency.

- Implicit and explicit dependencies:

Dependencies are implicit if those are only in the code within the class, and not outside the class or interfaces; while explicit dependencies exist between classes, and appear mostly in an object's constructor. Implicit class dependencies cost more to deal with, because they are more tightly coupled to other constructors, while explicit class dependencies are clear to identify their operational calls [39].

- Concrete and Abstract (Interface) dependencies:

If a class depends on an interface, then it does not depend on its concrete implementation, but some implementation, whereas the class cannot perform without its implementation. Developers would rather use the implementation of those interfaces than providing their own [40]. In Java, a concrete class is any class that can be directly created using a new operator; the class type is fixed when the code is compiled. A dependency occurs when one class utilizes another concrete class within its implementation. As experience in developing object-oriented systems has evolved, designs that minimize dependencies on concrete types have proven to be the most flexible. This flexibility is achieved through the use of abstract classes (like interfaces in Java).

- Direct and Indirect dependencies

If class A uses class B, then A is directly dependent on B. However if A depends on B and B depends on C, then A is indirectly dependent on C. In this empirical study, I use just direct dependencies.

In Tables 1–3, the possibilities of the three softwares for dependency computations are shown. Note that, we simply show the types of dependencies being used; one can explore these free softwares more or even add new measurement possibility to those already existing.

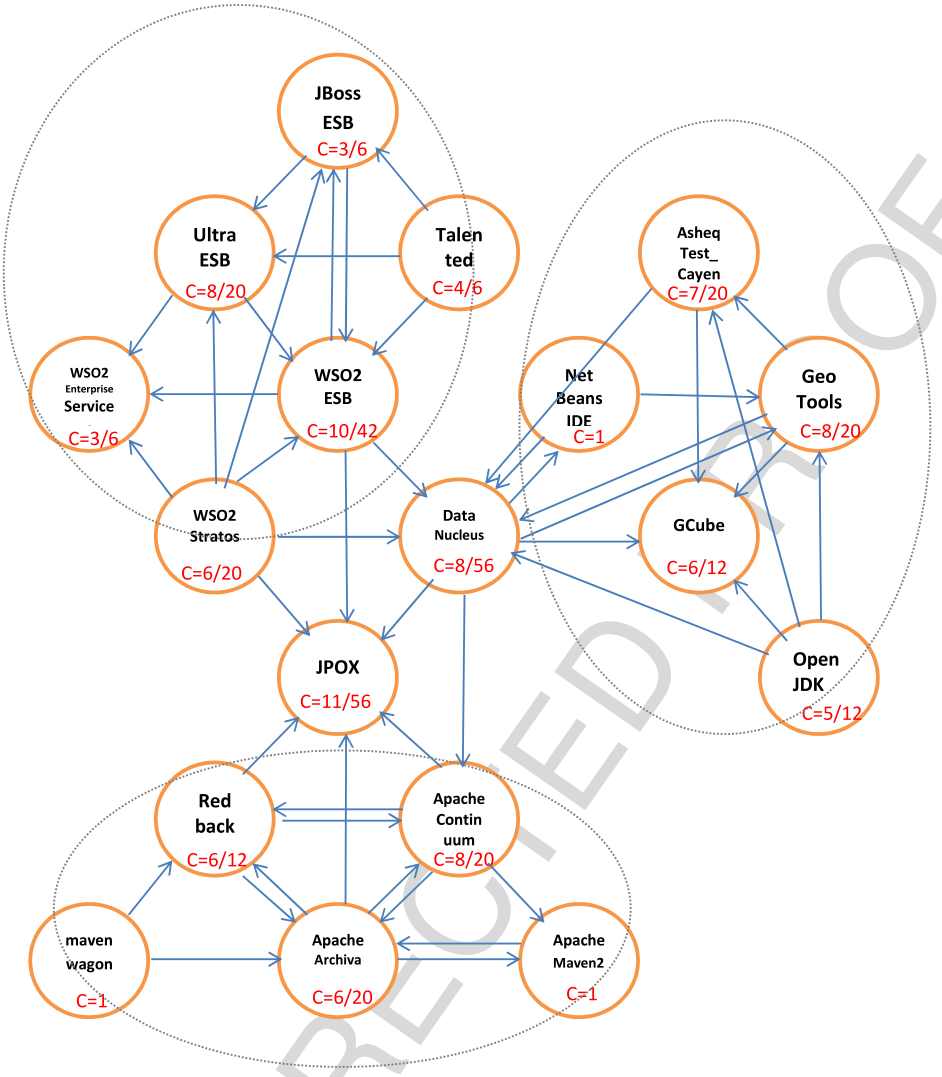

Fig. 3. Illustration of OSS reuse network divided into three clusters, each containing some OSS java projects. Connection between projects indicates the reuse, and numbers written in each project circle indicate clustering coefficient.

- All three softwares detect both concrete and abstract (or interface) classes.
- All three softwares render most classes and packages as *confirmed*, and others as *unconfirmed* [36,37]. Depfinder computes class dependencies for unconfirmed classes, but not Depanalyzer.
- Depfinder detects three types of explicit dependencies (class to class, method to class, method to method), as well as all implicit dependencies [36]; while, Depanalyzer and Jdepend detect just explicit dependencies.
- Most recent versions (snapshot) of the OSS projects shown in Figs. 2 and 3 are used. These include *maven 2, archiva, cumulus, dionysos, OAC, Jpox, netbeans, data nucleus, open jdk, emf, h2, WSO2 ESB, eclipse, Redback, WS02 Stratos, JBOSS ESB, Ultra ESB, Talented, Gcube, Geo tools*. They are downloadable from maven central repository, github, and java2s repository for jar files [41].

3.3. Empirical results

3.3.1. Empirical evaluation of sample OSS project network

In the empirical section, we considered sample network of open-source-software (OSS) projects reuse shown in Figs. 2 and 3 to identify the distinction between indegree and outdegree distribution. We obtained number of all project reuses from Ohloh website. Ohloh shows by which other projects, one is reused and how many times it is called. Unfortunately, Ohloh for each OSS project renders maximum of 100 inlinking projects' reuse as shown in Fig. 4 (left). However it is enough to show that in the double log diagram, indegree distribution represents indeed a power-law. We have also computed the

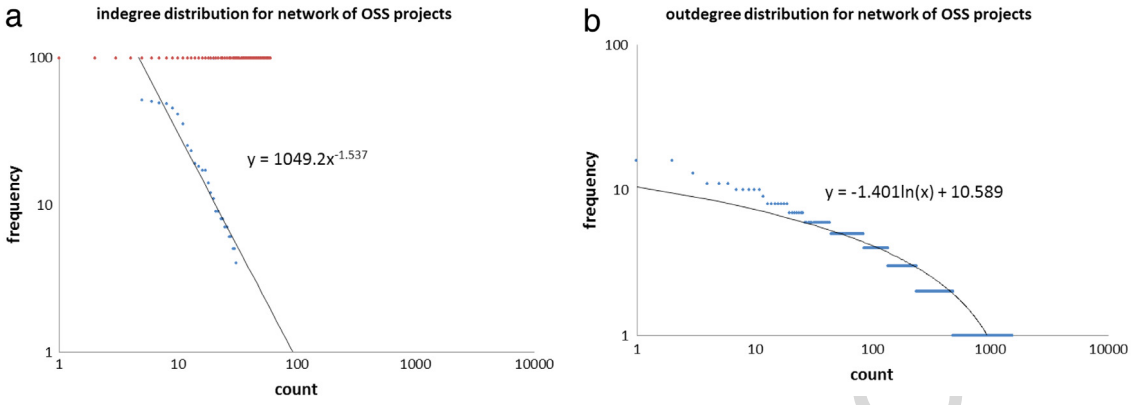

Fig. 4. Double Log degree distribution a. indegree distribution (left) b. outdegree distribution (right).

Table 1
Dependency type measurement possibilities.

Dep. Type	Class–class	Package–package	Package–class
Depfinder	Yes	No	Yes
Depanalyzer	Yes	Yes	Yes
Jdepend	No	Yes	Yes

Table 2
Dependency characteristics measurement possibilities.

Dep. Char.	Implicit/explicit	Concrete/abstract	Dir./inDir.
Depfinder	Both	Both	Direct
Depanalyzer	Explicit	Both	Direct
Jdepend	Explicit	Both	Direct

Table 3
Indegree/outdegree dependency measure possibilities.

Ind./Outd.	Class–class	Package–package	Package–class
Depfinder	Both	–	Both
Depanalyzer	Outdegree	Outdegree	Outdegree
Jdepend	–	Both	–

outdegree distribution in our sample of reuse network containing 1000 OSS projects, and depicted in the double log diagram in Fig. 4 (right). This clearly deviates from a power-law.

3.3.2. *Empirical evaluation of software project corpus*

In this section, first we evaluate the difference between outdegree/indegree class–class dependencies. Then the difference between outdegree/indegree package–package dependencies will be evaluated. To compute class–class dependencies, two softwares – Depfinder and Depanalyzer – are used for robust results. Both softwares count dependencies for both concrete classes and interfaces. Direct dependencies are only considered, although the softwares are able to distinguish between direct and indirect dependencies [37]. Depfinder considers both implicit and explicit dependencies, while Depanalyzer only considers explicit dependencies. As shown in Table 3, Depfinder can give us both indegree and outdegree class–class dependencies, which we use to check and confirm our results. We have downloaded the dependency graphs (metric) for different java projects, and computed the number of class dependencies using VB Excel. Then using un-normalized Complementary Cumulative Distribution Function (CCDF), we show the degree distribution for last version of java projects.

As one observes in Fig. 5 right, the outdegree class–class dependency obtained by Depfinder and Depanalyzer do not show heavy-tailed distribution; whereas indegree class–class dependencies (Fig. 5 left) feature a power-law distribution with exponential cut-off as shown also by Baxter [28]. This conforms to the intuition that one software could be indefinitely *reused (indegree dependency)*, so there tend to be many nodes with very small and very large degrees, therefore its indegree degree distribution is supposed to be heavy-tailed; while one software re*uses* limited number of others (*outdegree dependency*), so its outdegree distribution cannot be heavy-tailed.

Here, we try to evaluate the difference between outdegree/indegree of *package–package* dependencies. One can observe in Fig. 6 left and Fig. 6 right, both Jdepend and Depanalyzer softwares give similar package–package dependency, which is also a robustness test. As one sees in Fig. 6 left, indegree package–package dependencies feature a power-law distribution

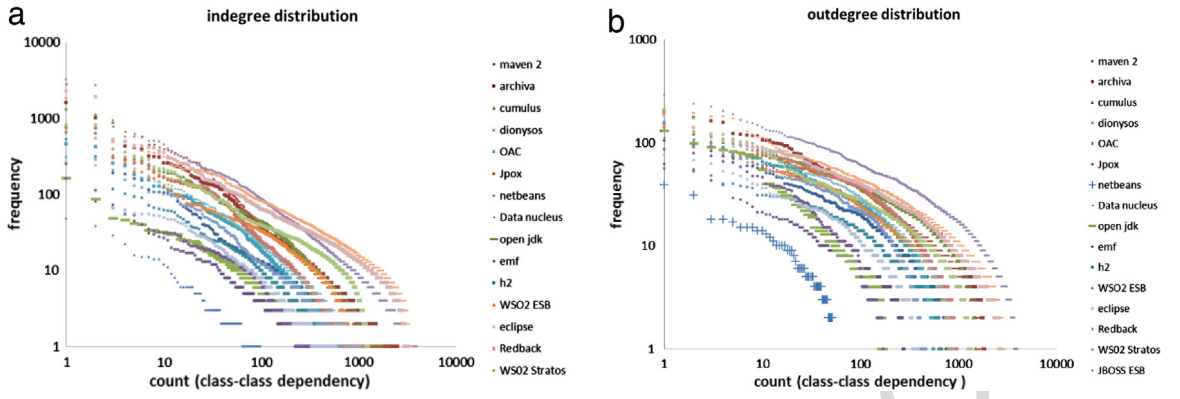

Fig. 5. Double Log class–class a. indegree dependency by Depfinder checked with Depanalyzer for robustness (left) b. outdegree dependency by Depfinder checked with Depanalyzer for robustness (right), for last snapshot of projects.

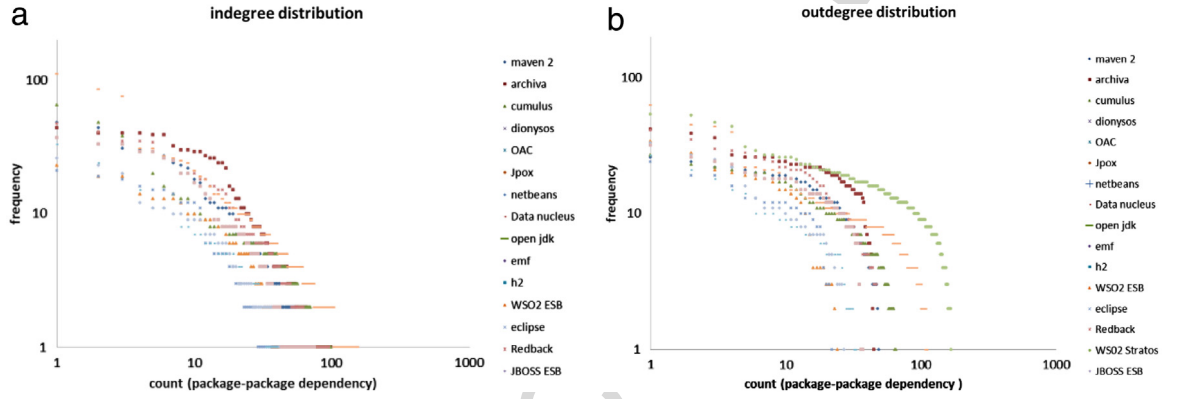

Fig. 6. Double Log package–package a. indegree dependency by Jdepend checked with Depanalyzer for robustness (left) b. outdegree dependency by Jdepend checked with Depanalyzer for robustness (right), for last snapshot of projects.

with flat top as shown also by Barker [27]; while, outdegree package–package dependencies obtained by both Jdepend and Depanalyzer do not show power-law distribution.

3.3.3. Fitting to data

In this subsection, we will fit the analytical models to our empirical results. Consider the degree distribution formula for indegree preferential attachment given in (6). In order to fit the model to the data, first set $m_0 = 0$.

$$p(k) = \frac{(m(1-\mu_1))^{\frac{1}{\mu_1}}}{\mu_1} k^{-\left(1+\frac{1}{\mu_1}\right)}. \tag{23}$$

Then fix m as average of indegree for each OSS project ($m = 50$ for class–class and $m = 6.5$ for package–package dependency in Archiva project), the only remaining parameter in the indegree distribution is μ_1. We should regress $P(k)$ over k to calculate $1/\mu_1$ for each OSS project. This results in $\mu_1 = 5$ for class–class and $\mu_1 = 1.67$ for package–package dependency in Archiva project. As observed in Fig. 7, we do see flat-top for indegree distributions as shown also by Barker [27], but we do not see exponential cut-off for indegree distribution as shown also by Baxter et al. [28]. Same procedure can be applied for outdegree distribution model in (19). Again in order to fit the model, set $m_0 = 0$; for large t we will have:

$$p(k) = \frac{1}{\mu_2} e^{\left(-\frac{k}{\mu_2}\right)}. \tag{24}$$

If we regress $p(k)$ for random attachment over k to calculate μ_2, it does not fit. It means that, this network is not based on random attachment. Then we use the fitness model as:

$$P(k) = \frac{x_{Max}^2}{N\langle x\rangle} f\left(\frac{x_{Max}^2}{N\langle x\rangle} k\right). \tag{25}$$

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035

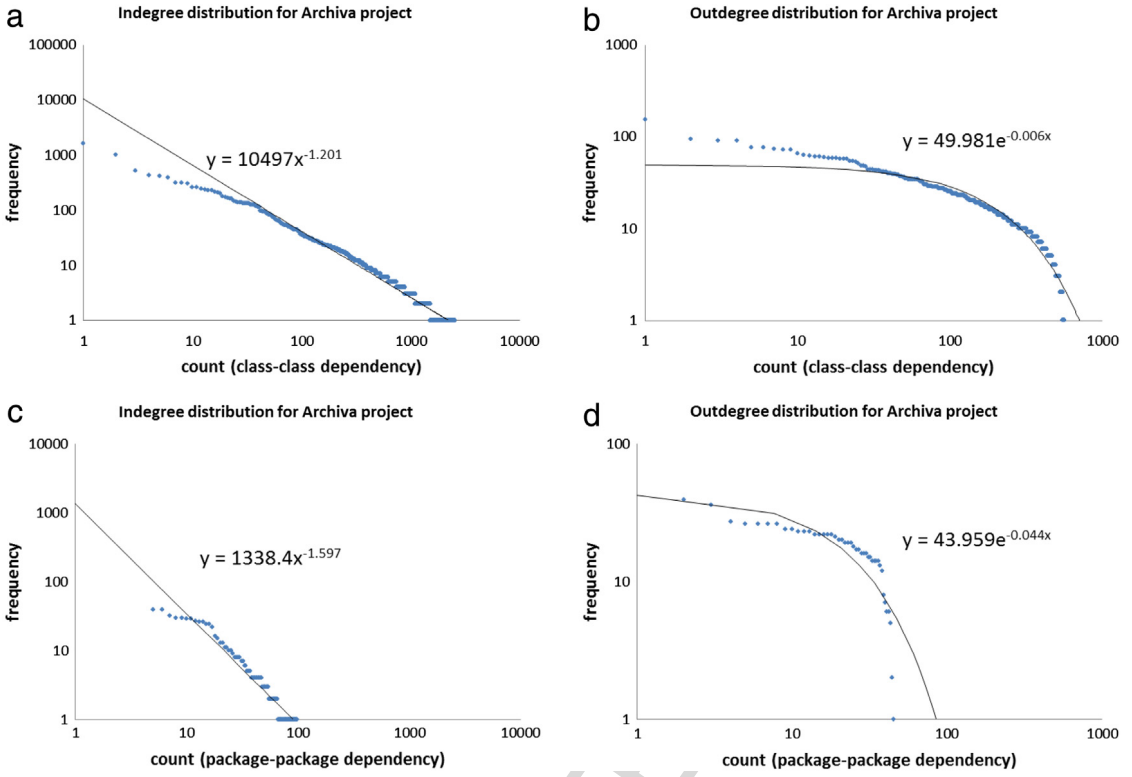

Fig. 7. Double Log degree distribution a. indegree for class–class dependency (top-left) b. outdegree for class–class dependency (top-right), c. indegree for package–package dependency (bottom-left), d. outdegree for package–package dependency (bottom-right).

Consider $f = e^{(-\frac{x^2}{N\langle x\rangle}\frac{k}{\mu_2})}$ as an exponential function, where $N\langle x\rangle$ as total number of nodes ($N\langle x\rangle = 2530$ for class–class and $N\langle x\rangle = 50$ for package–package dependency in Archiva project), then we regress $p(k)$ over k. OSS project outdegree distribution sound to be exponential as given by the formula for both class–class and package–package dependencies.

4. Conclusion

In this paper, first we discussed the importance of *directed* networks, where outlinks have been often neglected in other studies. Second, we analyzed *causal factors* for distinction between indegree and outdegree distributions (where this distinction has already been noticed in Barker et al. [27] and Baxter et al. [28]) for sample network of OSS projects as well as Java software corpus. Third, we investigated whether this distinction holds for different levels of system decomposition from project–project to package–package dependency and finally down to class–class dependency.

We emphasized the importance of studying indegree and outdegree distribution distinction, and why type of distribution is significant, in terms of (a) Structural property of complex network, (b) Statistical property of complex network, (c) Self-organizing property of complex network, (d) decomposability property of complex network.

Within undirected networks, we noted that scientists have shown much interest in presenting power-law features within complex networks, e.g. Barabási and Albert [3] claimed common property of many large networks in that the vertex connectivity follows scale-free power-law distribution, whereas authors such as Jiang et al. [18] discussed *deviation* from power-law distribution. Some others have also addressed this deviation and indicated size effect (Bagrow et al. [19]), information filtering mechanism (Mossa et al. [20]), and birth and death process (Shi et al. [21]) to account for this phenomena.

Within directed networks, most authors have either considered that outlinks follow similar mechanism of creation as inlinks' formation (Faloutsos et al. [17], Krapivsky et al. [24], Tanimoto [25]) in that link creation rate is the linear function of node degree, resulting in power-law shape for both indegree and outdegree distribution. At the same time, we claimed that (1) Outlinks feature different degree distributions from inlinks; and different link formation mechanisms cause the distribution distinctions, (2) in/outdegree distribution distinction holds for different levels of system decomposition; therefore this distribution distinction is a property of directed networks. We attempted to prove our claims both analytically and empirically. In the analytical section, apart from using indegree-based preferential attachment mechanism introduced by Barabási and Albert [3], we applied other link formation mechanisms such as outdegree-based preferential attachment, fitness-based preferential attachment, quality-based attachment and random attachment and assortativity.

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035

In the empirical section, we first considered sample network of open-source-software (OSS) projects reuse, where there was clear distinction between indegree and outdegree distribution; second we analyzed the network of java software corpus, where we noticed also clear distinction between indegree and outdegree distribution for both class–class and package–package dependencies. We concluded that indegree and outdegree do not lead to similar type of distributions. Finally, we fitted the analytical models to the empirical data, and resulted that indegree dependencies follow overall power-law distribution (power-law with flat-top), while outdegree dependencies do not follow heavy-tailed degree distribution (we obtained exponential fit to archiva open software project).

References

[1] Paul Erdős, A. Rényi, On the evolution of random graphs, Publ. Math. Inst. Hung. Acad. Sci. 5 (1960) 17–61.
[2] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (6684) (1998).
[3] A.L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (1999).
[4] G. Bianconi, A.L. Barabási, Competition and multiscaling in evolving networks, Europhys. Lett. 54 (2001) 436.
[5] S.N. Dorogovtesev, J.F.F. Mendes, A.N. Samukhin, Structure of growing networks with preferential linking, Phys. Rev. Lett. 85 (21) (2000) 4633.
[6] S.N. Dorogovtesev, J.F.F. Mendes, Effect of the accelerating growth of communications networks on their structure, Phys. Rev. E 63 (2) (2001) 025101.
[7] D.H. Shi, Q.H. Chen, L.M. Liu, Markov chain-based numerical method for degree distributions of growing networks, Phys. Rev. E 71 (3) (2005) 036140.
[8] Z. Liu, Y.C. Lai, N. Ye, P. Dasgupta, Connectivity distribution and attack tolerance of general networks with both preferential and random attachments, Phys. Lett. A 303 (5) (2002) 337–344.
[9] G. Ergun, G.J. Rodgers, Growing random networks with fitness, Physica A 303 (2002).
[10] X.J. Xu, L.M. Zhang, L.J. Zhang, Mutual selection in network evolution: The role of the intrinsic fitness, Internat. J. Modern Phys. C 21 (1) (2010).
[11] G. Caldarelli, A. Capocci, P.D.L. Rios, M.A. Munoz, Scale-free networks from varying vertex intrinsic fitness, Phys. Rev. Lett. 89 (2002) 258702. Q6
[12] M.E.J. Newman, Assortative mixing in networks, Phys. Rev. Lett. 89 (20) (2002) 208701.
[13] M. Piraveenan, K. Shing, K. Chung, S. Uddin, Assortativity of links in directed networks, in: FCS Conference, 2012.
[14] M. Piraveenan, M. Prokopenko, A.Y. Zomaya, Assortative mixing in directed biological networks, IEEE/ACM Trans. Comput. Biol. Bioinf. 9 (1) (2012) 66–78.
[15] M.O. Jackson, B.W. Rogers, Meeting strangers and friends of friends: how random are social networks? Amer. Econ. Rev. 97 (3) (2007) 890. Q7
[16] A.L. Barabási, H. Jeong, Z. Neda, E. Ravasz, A. Schubert, T. Vicsek, Evolution of the social network of scientific collaborations, Physica A 311 (2002) 590–614.
[17] M. Faloutsos, P. Faloutsos, C. Faloutsos, On power-law relationships of the Internet topology, in: SIGCOMM 99, 1999.
[18] J. Jiang, R. Wang, Q.A. Wang, Network model of deviation from power-law distribution in complex network, Eur. Phys. J. B 79 (2011) 29–33.
[19] J.P. Bagrow, J. Sun, D. Ben-Avraham, Phase transition in the rich-get-richer mechanism due to finite-size effects, J. Phys. A 41 (2008) 185001.
[20] S. Mossa, M. Barthelemy, H.E. Stanley, L.A.N. Amaral, Truncation of power law behavior in "scale-free" network models due to information filtering, Phys. Rev. Lett. 88 (2002) 138701.
[21] D. Shi, M. Liu, X. Zho, H. Zhou, B. Wang, Evolving networks and birth-and-death processes, 2005. http://arxiv.org/abs/math-ph/0506019.
[22] T. Maillart, D. Sornette, S. Spaeth, G. Von Krogh, Empirical tests of Zipf's law mechanism in open source Linux distribution, Phys. Rev. Lett. 101 (2008) 218701.
[23] T. Maillart, D. Sornette, S. Frei, T. Duebendorfer, A. Saichev, Quantification of deviations from rationality with heavy tails in human dynamics, Phys. Rev. E 83 (5) (2011) 056101.
[24] P.L. Krapivsky, G.J. Rodgers, S. Redner, Degree distribution of growing networks, Physica A (2001).
[25] S. Tanimoto, Power laws of the in-degree and out-degree distributions of complex networks, Physica A (2009).
[26] Z.Q. Jiang, W.X. Zhou, Complex stock trading network among investors, Physica A (2010).
[27] R. Barker, B.E. Carpenter, E. Tempero, Deviations from power law in citation distributions' computer science Technical Reports, 2010, ISSN 1173-3500.
[28] G. Baxter, et al., Understanding the shape of java software, in: OOPSLA'06, ACM, 2006. Q8
[29] H. Simon, The organization of complex systems, in: H.H. Pattee (Ed.), Hierarchy Theory: The Challenge of Complex Systems, George Braziller, New York, NY, 1973.
[30] A.L. Barabási, R. Albert, H. Jeong, Mean-field theory for scale-free random networks, Physica A 272 (2008).
[31] R. Harrison, L.G. Samaraweera, M.R. Dobie, P.H. Lewis, Comparing programming paradigms: an evaluation of functional and object-oriented programs, Softw. Eng. J. 11 (1996) 247–254.
[32] Y.H. Wang, C.M. Chung, T.K. Shih, H.C. Keh, J.F. Chen, The complexity measurement of software through program decomposition, Comput. Syst. Sci. Eng. 15 (2000) 127–134.
[33] J. Ma, D. Zeng, H. Zhao, Modeling the growth of complex software function dependency networks, Inf. Syst. Front. 14 (2012) 301–315.
[34] L. Wen, D. Kirk, R.G. Dromey, Proc. 6th IEEE Int. Conf. on Cognitive Informatics, ICCI'07, 2007.
[35] J. Ruths, D. Ruths, Control profiles of complex networks, Science 343 (6177) (2014) 1373–1376.
[36] dependency finder: http://depfind.sourceforge.net/.
[37] dependency analyzer: http://www.dependency-analyzer.org/.
[38] jdepend: http://clarkware.com/software/JDepend.html.
[39] explicit dependencies principal: http://deviq.com/explicit-dependencies-principle.
[40] Understanding dependencies tutorial: http://tutorials.jenkov.com/ood/understanding-dependencies.html.
[41] maven repository: http://search.maven.org, github repository: https://github.com, jar file repository: http://java2s.com.

Please cite this article in press as: S.K. Behfar, et al., Directed networks' different link formation mechanisms causing degree distribution distinction, Physica A (2016), http://dx.doi.org/10.1016/j.physa.2016.06.035